Project 2: Pattern Matching in Compressed DNA Sequence

Barna Saha
email:barna@umd.edu

April 22, 2008

Abstract

Space efficient storage of large genome sequences requires good compression techniques. However, if these sequences need to be decompressed, before any processing can be done over them, the advantage of compression is lost. New techniques are required to extend the traditional pattern matching algorithms to work directly on the compressed sequence. This saves space in memory, requires less disk access and results in high speed up. In this project we will explore one such pattern matching algorithm on compressed DNA sequence, known as *Derivative Boyer-Moore* algorithm [2]. We will compare its running time with the traditional exact string matching algorithm, the *Boyer-Moore algorithm*[6], the fastest known exact string matching algorithm *AGREP* [3] and *LZgrep*, which is another algorithm that searches directly on the compressed sequence.

1 Project Description

Perhaps one of the most recurrent subproblems appearing in almost every applications of computer science is the need to find the occurrences of a pattern string inside a large text. The problem is especially important in computational biology, where large sized DNA sequences are searched for finding matching patterns. Each DNA sequence contains only four alphabets A,C,T,G, but they are generally very large in size and contains vast amount of information. The human genome for instance contains three billions characters over twenty-three pairs of chromosomes. Pattern matching over such long DNA sequences requires algorithms which can handle the sequences efficiently and have very fast time complexity for search operation. In order to save storage space, it is natural to store the DNA sequences in a compressed form in a secondary storage. However decompressing them in the main memory before searching, may result in memory overflow, multiple disk access and slower running time. To overcome these adverse effects, recently there is a surge of interest in designing pattern matching algorithms which look for exact occurrences of a pattern in a compressed DNA sequence without first decompressing it. The technique allows reduction

in the size of the DNA sequence and I/O overhead considerably and thus results in very fast searching time.

The compressed pattern matching problem was first defined in the work of Amir and Benson [1] as the task of finding pattern occurrences in compressed sequence without first decompressing it. Using variations of compression algorithms and search techniques, different algorithms like *Fast matching with encoded DNA sequences* (FED) [7], *Boyer Moore on byte pair encoding* (BB) [4], *Boyer Moore on Lempel-Ziv Compressed sequences* (LZ-Grep) [8], *Super-alphabet shift-or* (SASO) [5], and *Derivative Boyer Moore* (d-BM) [2] have been developed in the last few years.

In this project, we will explore the details of d-BM method, by implementing it and comparing it with the traditional *Boyer Moore*, *AGREP* and *LZGrep*. d-BM utilizes the fact that DNA sequence contains only four alphabets A, C, T, G. Thus text and pattern can be broken into segments of 4 characters, each of which is encodable in a single byte. This guarantees a compression ratio of 75%. In addition to this, such encoding increases the number of possible characters from 4 to 256 and results in larger shift defined by bad character rule and good suffix rule of *Boyer Moore algorithm*. As a future goal it will be interesting to extend this idea, for matching multiple patterns over compressed text and to allow inexact matching.

References

[1] A. Amir, and G. Benson, "Effcient two-dimensional compressed matching", In Proc. DCC'92, pp. 279-288, 2002.

[2] Chen, L., Lu, S., Ram, J., "Compressed Pattern Matching in DNA Sequences." In: CSB 2004. IEEE Computational Systems Bioinformatics Conference, pp. 6268 (2004).

[3] Wu, S., Manber, U.: "Fast Text Searching Allowing Errors." Communications of the ACM 35(10), 8391 (1992).

[4] Shibata, Y., Matsumoto, T., Takeda, M., Shinohara, A., Arikawa, S. "A Boyer-Moore Type Algorithm for Compressed Pattern Matching." In: Giancarlo, R., Sankoff, D. (eds.) CPM 2000. LNCS, vol. 1848, pp. 181194. Springer, Heidelberg (2000).

[5] Fredriksson, K. "Shift-Or String Matching with Super-Alphabets." Information Processing Letters 87(4), 201204 (2003).

[6] Boyer, R.S., Strother Moore, J. "A Fast String Searching Algorithm." Communications of the ACM 20(10), 762772 (1977).

[7] Jin Wook Kim, Eunsang Kim, and Kunsoo Park, "Fast Matching Method for DNA Sequences", Book of Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, 271-281, 2007.

[8] Gonzalo Navarro and Jorma Tarhio, “LZgrep: a BoyerMoore string matching tool for ZivLempel compressed text: Research Articles”, Softw. Pract. Exper., 35, 2005.